Introduction to Sybase in IR1

Carol Bonney

520-TD-001-002   SSI&T 11-1

Discussion Topics

This briefing is designed to give you only an overview of Sybase and the PDPS database for IR1. More formal training will be provided by the Sybase vendor at a later date. This overview consists of:
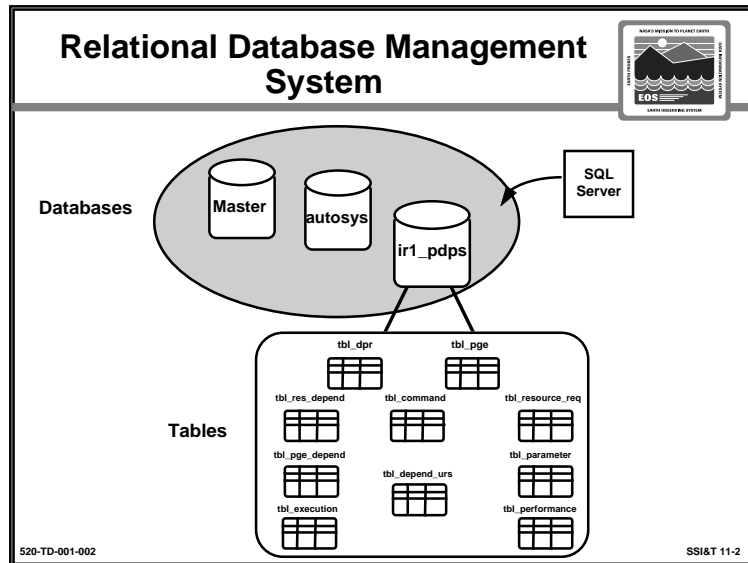
- basic structure and fundamentals of Sybase

- ir1_pdps database tables

- logging into ir1_pdps

- isql and scripts

- system procedures

- custom scripts

- transaction log

- error log

- database administration

References:

SA Companion User's Guide (Sybase), 10.0.1, Sybase Inc., 1994

System and Database Administration, Student Guide, Volumes 1 & 2, Sybase Inc., 1993

Fast Track to Sybase, Student Guide, Volumes 1 & 2, Version 2.1, Sybase Inc, 1994

Relational Database Management System

Databases: Master, autosys, ir1_pdps — SQL Server

Tables: tbl_dpr, tbl_pge, tbl_res_depend, tbl_command, tbl_resource_req, tbl_pge_depend, tbl_depend_urs, tbl_parameter, tbl_execution, tbl_performance

520-TD-001-002   SSI&T 11-2

Discussion Topics

Logging into the SQL Server will provide access to a number of databases (e.g. IR1 PDPS database, AutoSys database, Master database). Within each database are a series of tables which contain data.. For example, the IR1 PDPS database contains 10 different tables.

These tables are related to each other in various ways:

- one-to-one

- one-to-many

- many-to-many

More information on the IR1 PDPS database will be provided in a moment.

Sybase Tables

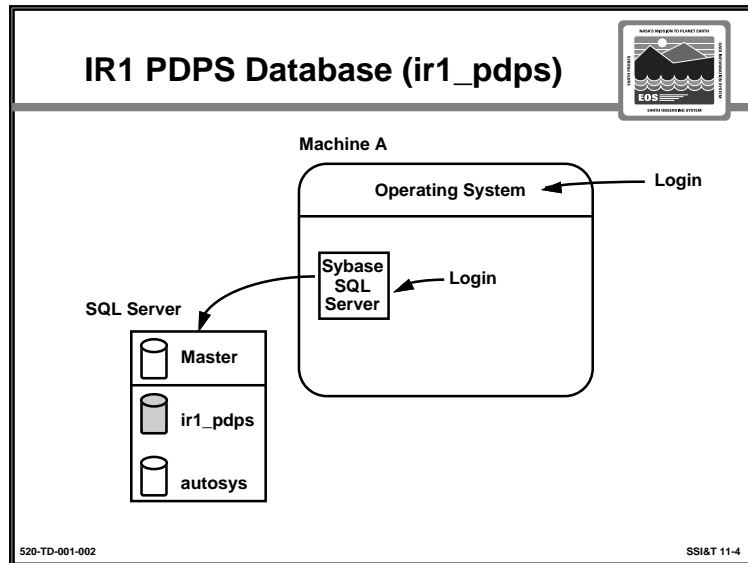- **Tables**
  - **Column Name**
  - **Datatype**
  - **Column Property**

Column Name	Datatype	Column Property
dpr_id	char(36)	NOT NULL
pdps_version	char(32)	default "NONE"
creation_date	datetime	NOT NULL

520-TD-001-002   SSI&T 11-3

Discussion Topics

- Tables contain columns which consist of:

  - column name,

  - system or user-defined datatype (e.g., type of data, size, and storage format of columns), and

  - column property (e.g., rules, and defaults). If a data value is "null", the value is unknown; "not null" is the SQL system default column property
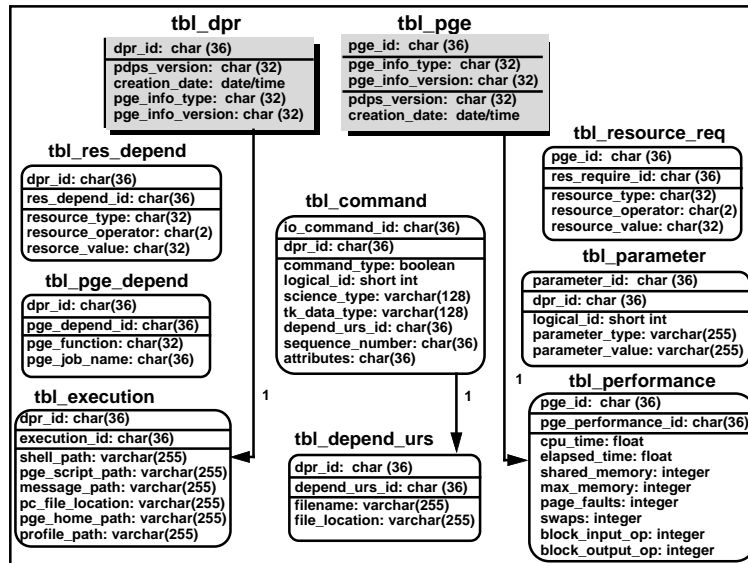
  (*Fast Track to SYBASE, Student Guide, Vol. 1, p. 5-2, 5-10, Ver 2.1, (1994), Sybase, Inc.*)

- This example shows the dpr_id table from the IR1 PDPS database

IR1 PDPS Database (ir1_pdps)

Machine A

Operating System ← Login

Sybase SQL Server ← Login

SQL Server

Master

ir1_pdps

autosys

520-TD-001-002   SSI&T 11-4

Discussion Topics

Sybase is the "engine" of the system and is located at each of the DAACs on the database server.   As stated previously, both the autosys and IR1_pdps databases are managed by a single Sybase SQL Server. A Sybase SQL Server is a set of one or more cooperating processes that manage one or more databases and provide database access to multiple users.
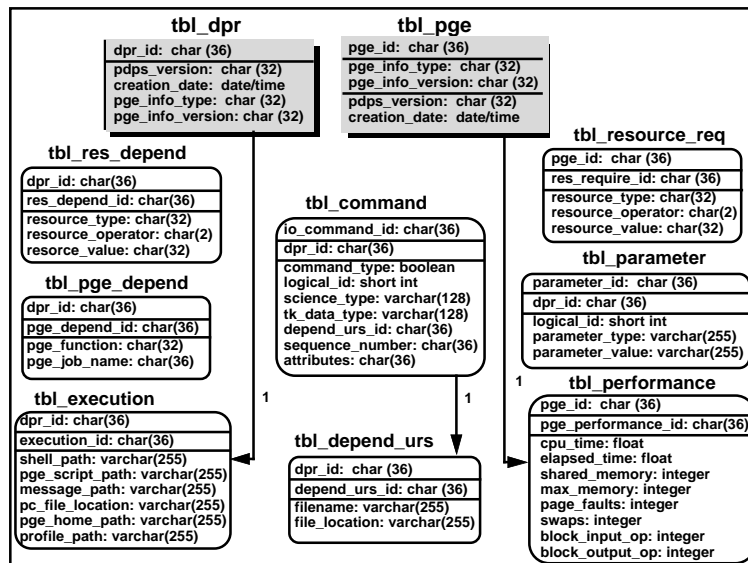
- Typically, once you have logged into the SQL Server, you will automatically be connected to the "*Master*" database. *Master* contains system tables and information about who can connect to the Server. These tables contain information for the overall configuration of the database server and all database objects associated with the server.

- The user can change this default to pdps or autosys database by typing:
  ```
  sp_modifylogin login_name, option, value
  (e.g. sp_modifylogin ir1_trng, defdb, ir_pdps)
  ```

tbl_dpr

| dpr_id: char (36) |
| pdps_version: char (32) |
| creation_date: date/time |
| pge_info_type: char (32) |
| pge_info_version: char (32) |

tbl_pge

| pge_id: char (36) |
| pge_info_type: char (32) |
| pge_info_version: char (32) |
| pdps_version: char (32) |
| creation_date: date/time |

tbl_res_depend

| dpr_id: char(36) |
| res_depend_id: char(36) |
| resource_type: char(32) |
| resource_operator: char(2) |
| resorce_value: char(32) |

tbl_command

| io_command_id: char(36) |
| dpr_id: char(36) |
| command_type: boolean |
| logical_id: short int |
| science_type: varchar(128) |
| tk_data_type: varchar(128) |
| depend_urs_id: char(36) |
| sequence_number: char(36) |
| attributes: char(36) |

tbl_resource_req

| pge_id: char (36) |
| res_require_id: char (36) |
| resource_type: char(32) |
| resource_operator: char(2) |
| resource_value: char(32) |

tbl_parameter

| parameter_id: char (36) |
| dpr_id: char (36) |
| logical_id: short int |
| parameter_type: varchar(255) |
| parameter_value: varchar(255) |

tbl_pge_depend

| dpr_id: char(36) |
| pge_depend_id: char(36) |
| pge_function: char(32) |
| pge_job_name: char(36) |

tbl_execution

| dpr_id: char(36) |
| execution_id: char(36) |
| shell_path: varchar(255) |
| pge_script_path: varchar(255) |
| message_path: varchar(255) |
| pc_file_location: varchar(255) |
| pge_home_path: varchar(255) |
| profile_path: varchar(255) |

tbl_depend_urs

| dpr_id: char (36) |
| depend_urs_id: char (36) |
| filename: varchar(255) |
| file_location: varchar(255) |

tbl_performance

| pge_id: char (36) |
| pge_performance_id: char(36) |
| cpu_time: float |
| elapsed_time: float |
| shared_memory: integer |
| max_memory: integer |
| page_faults: integer |
| swaps: integer |
| block_input_op: integer |
| block_output_op: integer |

Discussion Topics

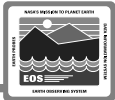**ir1_pdps database schema**. The ir1_pdps database consists of ten tables holding related info.

- **tbl_dpr** - This table stores information about each Data Processing Request (DPR) and has connections to tbl_command, tbl_parameter, tbl_res_depend, tbl_execution, and tbl_depend_urs. When the DPR request is made, the DPR table is created and a copy of members belonging to a selected PGE from each of these tables is made and attached to requested DPR. The DPR table itself contains:

  – pdps_version   the version of the software

  – creation_date   date created

  – pge_info_type   the pge that was selected for the DPR

  – pge_info_version   the version of the PGE

- **tbl_pge** - This table is exactly the same as the DPR table except that it serves the initial registration of the PGE into the data base from the PCF file

- **tbl_res_depend** - This table holds information on any resource dependencies that are applicable to PGE and DPR. For example, machine = dps1sgiedf states that the PGE should be run on dps1sgiedf. It contains three members:

  – resource_type   e.g. machine

  – resource_operator   e.g. =

  – resource_value   e.g. dps1sgiedf

  – Note: In the current version only the machine name is supported. Later this can expanded to include disk size etc.

- **tbl_pge_depend** - This table hold information on dependencies that a current DPR might have on previous dependencies and provides a way to link the execution of DPRs. For example, one would like to run a particular PGE only if a certain other PGE has run successfully. The table contains:

  – pge_function   SUCCESS or FAILURE

  – pge_job_name   name of the PGE

tbl_dpr
dpr_id: char (36)
pdps_version: char (32)
creation_date: date/time
pge_info_type: char (32)
pge_info_version: char (32)

tbl_pge
pge_id: char (36)
pge_info_type: char (32)
pge_info_version: char (32)
pdps_version: char (32)
creation_date: date/time

tbl_resource_req
pge_id: char (36)
res_require_id: char (36)
resource_type: char(32)
resource_operator: char(2)
resource_value: char(32)

tbl_res_depend
dpr_id: char(36)
res_depend_id: char(36)
resource_type: char(32)
resource_operator: char(2)
resorce_value: char(32)

tbl_command
io_command_id: char(36)
dpr_id: char(36)
command_type: boolean
logical_id: short int
science_type: varchar(128)
tk_data_type: varchar(128)
depend_urs_id: char(36)
sequence_number: char(36)
attributes: char(36)

tbl_parameter
parameter_id: char (36)
dpr_id: char (36)
logical_id: short int
parameter_type: varchar(255)
parameter_value: varchar(255)

tbl_pge_depend
dpr_id: char(36)
pge_depend_id: char(36)
pge_function: char(32)
pge_job_name: char(36)

tbl_performance
pge_id: char (36)
pge_performance_id: char(36)
cpu_time: float
elapsed_time: float
shared_memory: integer
max_memory: integer
page_faults: integer
swaps: integer
block_input_op: integer
block_output_op: integer

tbl_execution
dpr_id: char(36)
execution_id: char(36)
shell_path: varchar(255)
pge_script_path: varchar(255)
message_path: varchar(255)
pc_file_location: varchar(255)
pge_home_path: varchar(255)
profile_path: varchar(255)

tbl_depend_urs
dpr_id: char (36)
depend_urs_id: char (36)
filename: varchar(255)
file_location: varchar(255)

Discussion Topics

- **tbl_execution** - This table hold information on some of the variables that are needed to run the PGE. User should be concerned with only three:
    - pge_script_path   full path to the PGE executable
    - pc_file_location   full path to the PCF file that gets generated
    - profile_path   full path to the resource file that gets written after the pge execution
- **tbl_command** - This table contains information on all the file descriptions in the PCF file. Note: the dpr_id field refers to both pge_id in tbl_pge and the dpr_id in tbl_dpr. The user should only be concerned with the following members:
    - logical_id   The logical ID of the file science type (e.g., MODIS)
    - tk_data_type   There are about six different types of files in the PCF eg. PRODUCT_INPUT_FILE
    - depend_ur_id   user does not directly relate to this parameter but it acts as a pointer to file mappings in the UR table. File mappings are the directory path and the file name.
    - attributes   There is a file attribute field in the PCF where user can put some description about the file.
- **tbl_depend_urs** - As mentioned above, this table has a one to one mapping with the commands table. Note: the dpr_id field refers to both pge_id in tbl_pge and the dpr_id in tbl_dpr. This table contains:
    - filename   name of the file in the PCF file_location path to the file
- **tbl_resource_req** - the general resource requirements for a PGE type (i.e., part of the PGE Profile), as opposed to the tbl_res_depend which is the resources required by an individual instance (i.e., a run) of a PGE.
- **tbl_parameter** - In PCF there is a section where user can define some parameters of the format name = value. This table therefore contains:
    - logical ID   the ID of the parameter defined in the PCF
    - parameter_type   name of the parameter
    - parameter_value   value of the parameter eg. 10220 HOSTNAME monet
- **tbl_performance** - This table is intended for Release A (i.e for future use)

**Fundamental Database
Environment Variables**

- **SYBASE - specifies the path name of the Sybase
  directory**
- **DSQUERY - specifies the name of the Sybase SQL
  Server for all client applications.**

520-TD-001-002   SSI&T 11-7
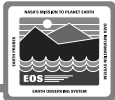
Discussion Topics

Fundamental database environment variables:

- **SYBASE** - This specifies the path name of the Sybase directory

  - The Sybase directory contains all files associated with Sybase configuration and operational files.

  - The interfaces file must reside on the Database Server, the (other) SSI&T Workstation (Sun), and the SSI&T Processor (SGI). Each entry in the file tells the host machine how to connect to a SQL Server. This file is normally located in the Sybase directory.

- **DSQUERY** - This specifies the name of the Sybase SQL Server for all client applications.

Setting Environment Variables

```
> login
> password
> setenv SYBASE/vendor/Sybase
> setenv DSQUERY nickalus_srvr
> cd $SYBASE/bin

> set path = ($PATH $SYBASE/bin)
```

520-TD-001-002   SSI&T 11-8

Discussion Topics

Before you are able to access Sybase, you need to set your environmental variables. These variables can be customized by the user

Exercise: Login and set your environmental variables

- Type x host + and machines IP address on local machine (command tool)

- Login to the Sun (dps3sunedf): telnet 192.150.28.116

- Password

- setenv for SYBASE and DSQUERY

- include $SYBASE/bin in your path

- If you need to set path = ($PATH $SYBASE/bin)

Logging into Ir1 PDPS Database

- **Login: `isql -Uir1_trng`**
- **Password: `ir1_trng`**
  ```
  1> use ir1_pdps
  2> select dpr_id from tbl_execution
  3> go
  ```

520-TD-001-002   SSI&T 11-9

Discussion Topics

Accessing SQL Server with the isql Client. It has the following features:
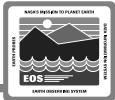
- "isql is an 'interactive SQL' utility.

- handles the connection to SQL Server

- accepts input from the keyboard or input from a file containing Transact-SQL commands

- sends ASCII text to the server

- displays the results fo the query (returned rows and messages) on screen, with basic formatting." (*Fast Track to SYBASE, Student Guide, Vol. 1, p. 3-4, Ver 2.1, (1994), Sybase, Inc.*)

To access Sybase, you must login by typing:

    – **`isql -Uir1_trng`**

- Password:

    – **`ir1_trng`**

- Once you have logged in, you are automatically in your default database. If you want to access another database, type in **use *database_name***:

    – **`use ir1_pdps`**   (note: ir1_pdps is the default for the ir1_trng account)

- To select data from the database tables, use the select command (for example):

    – **`select dpr_id from tbl_execution`**

    – **`go`**

Exercise: direct class to use select command to bring up ir1_pdps tables

---

Useful System Procedures

- **Stored Procedures**
  - `> sp_help [object_name]`
  - `> sp_helpdb [object_name]`
  - `> sp_helptext [object_name]`
  - `> sp_helpindex[object_name]`
  - `> sp_who`
  - `> sp_adduser          (*sa only)`
  - `> sp_addlogin         (*sa only)`
  - `> sp_modifylogin`
  - `> sp_helpdevice [device_name]`
  - `> sp_helpuser [user_name]`

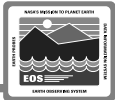520-TD-001-002   SSI&T 11-10

Discussion Topics

Stored procedures are SQL statements that provide a method of accessing and manipulating the data stored in the tables. For example,

- the "`sp_help`" reports information about the specified table such as the field names within the tables.

- "`sp_helpdb`" displays the size, owner, database ID, creation date, and option settings for a particular database or for all databases

- "`sp_helptext` *view_name* "reports the text used to create a stored procedure, trigger, etc.

- "`sp_helpindex`" provides index information on a table

- "`sp_who`" displays currently logged on users

- "`sp_adduser`" adds a user (only sa can do this)

- "`sp_addlogin`" adds a user login (only sa can do this)

- "`sp_modifylogin`" allows you to modify user login

- "`sp_helpdevice [device_name]`" displays information about a specified device

- "`sp_helpuser [user_name]`" displays information about a specified user

(*Fast Track to SYBASE, Student Guide, Vol. 2, p. Ch 11, Ver 2.1, (1994), Sybase, Inc.*)
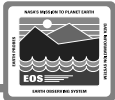
Custom Scripts for Users

- **Viewing DPRs and PGEs**
  - > `sp_view_dpr`
  - > `sp_view_pge`
- **Deleting DPRs and PGEs**
  - > `sp_delete_dprs`
  - > `sp_delete_pges`

520-TD-001-002   SSI&T 11-11

Discussion Topics

- To view a Data Processing Requests (DPR) or PGE,
  - **`sp_view_dpr`** - procedure to view a (specific) dpr and all related information from child tables
  - **`sp_view_pge`** - procedure to view a pge and all related info. contained in child tables
- The accumulation of DPRs in the PGE database can eventually lead to the database becoming full.
  - The custom Sybase stored procedure, **`sp_delete_dprs`** is used to delete unwanted DPRs and all related information from child tables.
- If a PGE needs to be deleted,
  - **`sp_delete_pges`** - procedure to delete a pge and all related information from child tables (note: this procedure should be performed carefully, and will probably not occur frequently)

Transaction Log

- **Each SQL Server database has its own transaction log**
- **System table =** *syslogs*
  - **managed exclusively by SQL Server**
  - **contains a record of changes made in the database in the order in which they occur**
  - **SA can check how full syslogs is:**
    >dbcc checktable (syslogs)
- **Any transaction may generate a corresponding insertion into the** *syslogs* **table for the affected database**
- **Transaction Log maintenance (automatic)**
  - **> sp_interimthresholdaction**
  - **> sp_thresholdaction**

520-TD-001-002   SSI&T 11-12

Discussion Topics

A transaction log is updated for all transactions that occur within the database. This log is used to perform a complete database recovery in the event of a media failure.

- the start and end of transactions are recorded

- before and after images of all data modifications are recorded

- *syslogs* is a table in the SQL Server database (*Master*) that is shared by all users

- when a transaction is written to *syslogs*, SQL Server guarantees that the transaction can be recovered

(*Fast Track to SYBASE, Student Guide, Vol. 1, p. 9-6, Ver 2.1, (1994), Sybase, Inc.*)

- The System Administrator has the ability to check how full the syslogs are by typing:
  `dbcc checktable (syslogs)`

- **Transaction Log Maintenance.** Since the transaction log records all changes to its database, it grows in size over time, and can cause problems when full. The transaction log must be dumped periodically. Custom scripts have been written to clear the transaction logs for the PDPS databases automatically:
  - **sp_interimthresholdaction** - Clears out the PDPS Sybase transaction log when it is approximately 50% full.
  - **sp_thresholdaction** - Clears out the PDPS Sybase transaction log when it is close to stopping the Sybase server.

Error Log

- **To access the Sybase error log, type (from UNIX):**
  ```
  > cd/$SYBASE/install
  > ls
  ```

```
┌─────────────────────────────────────────────────────────────┐
│  ▽            cmdtool (CONSOLE) – /bin/csh                   │
├─────────────────────────────────────────────────────────────┤
│ dps3sunedf:/data/sybase/install>ls                          │
│ Nerrorlog               db.dump              showserver      │
│ RUN_nickalus_srvr       dump.sql             startserver     │
│ RUN_nickalus_srvr_backup sac.log             sybinit         │
│ SPR                     setperm_all          sybinit.server  │
│ backup.log              setperm_monserv      us_english      │
└─────────────────────────────────────────────────────────────┘
```

520-TD-001-002   SSI&T 11-13

Discussion Topics

Sybase errors are logged at the system console and on the SQL Server error log.   To display the errorlog, type: cd/$SYBASE/install

- the Sybase error log and the custom "Event Logger" are separate items.

- Sybase does, however, serve as the database for the "Event Logger", but not in real-time.

Using isql Client with Prepared ASCII Files

Sample Script

```
create procedure sp_view_dpr
    (@dprId          char(36))
as

declare @run_date           datetime

declare @dpr_pdps_version   char(32)
declare @dpr_creation_date  datetime
decalre @dpr_pge_info_type  char(32)
declare @dpr_pge_info_version        char(32)

select @run_date=getdate()
/*  DPR TABLE */

select dpr.pdps_ver    pdps    sion
```

- **To read in an ASCII file containing a query or queries (script) for execution by isql, type:**

    ```
    isql -U login_name -P password -i file_name
    ```
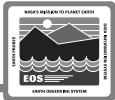
520-TD-001-002   SSI&T 11-14

Discussion Topics

- Script files are text files containing one or more batches of Transact-SQL code, each terminated by "go"

- Scripts can be used to create or alter databases, add logins and users, create objects, set permissions, load small amounts of data, etc.

- To read in an ASCII file containing a query or queries (script) for execution by isql, type:

− `isql -U login_name -P password -i file_name`

Database Administration (M&O)

- **Tasks**
  - **Database backup**
    `dump database` *`database_name`* `to` *`dump_device`*
  - **Database restoral**
    `load database` *`database_name`* `from` *`dump_device`*
  - **Startup and shutdown of the SQL Server**
  - **Transaction Log maintenance**
  - **Adding users to the database**
  - **Adding logons**
  - **Deleting users from the database**
  - **Grant/revoke permissions**
- **Roles**
  - **System Administrator (sa)**
  - **System Security Officer (SSO)**
  - **Operator**
  - **Database Owner (DBO)**

520-TD-001-002   SSI&T 11-15

Discussion Topics

The Database Administrator is responsible for the overall health of the database. This person can restrict database access through database privileges and can enforce security through auditing the database. In order to perform any database administration, you must have granted permissions access to do so.

- The operations staff are responsible for performing database backups. Database backup is required for two reasons. First, the database must be backed up on a regular basis in order to protect the data from loss due to system failure. Second, database backup must be done in order to preserve the database configuration set up by one instrument team, prior to turning the system over to another instrument team.
  - **Database backup** is accomplished with the **dump database** command. This should be done when the database is in single-user mode or is not in use.
  - **Database restoration** is accomplished with the **load database** command.
- Other server/database administration procedures can be found in the System Administration Manual and the SYBASE SQL Server Reference Manual Vols 1&2. These procedures include:
  - Startup (`$SYBASE/install`) and shutdown of the SQL Server
  - Transaction Log Maintenance
  - Adding Users to the database
  - Adding logins (account)
  - Deleting users from the database
  - Grant/revoke permissions
- Different Sybase Roles (see Chapter 2 in the System Administrator's Guide for details). These roles are not currently set up in the IR1 PDPS database -- it is up to the sites to determine if they want to set up these roles.

Sybooks

> `$SYBASE/sybooks`

- **Two environment variables:**
  - `setenv SYBROOT $SYBASE/sybooks`
  - `setenv EBTRC $SYBROOT/sun5m/.ebtrc`

520-TD-001-002   SSI&T 11-16

Discussion Topics

How to set environment variables and gain access to Sybase on-line documentation